08-02-00

A

Jc806 U.S. PTO
08/01/00

jc784 U.S. PTO
09/629337
08/01/00

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))
Attorney Docket No. 0100.0001160 Total Pages 32
First Inventor or Application Identifier Mark C. Fowler
Title Optimized Primitive Filler
Express Mail Label No. EL504284757US

APPLICATION ELEMENTS *See MPEP chapter 600 concerning utility patent application contents.*	*ADDRESS TO:* **Assistant Commissioner for Patents Box Patent Application Washington, DC 20231**

1. ☒ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)
2. ☒ Specification *Total Pages* 22
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to Microfiche Appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings *(if filed)*
- Detailed Description
- Claim(s)
- Abstract of the Disclosure
3. ☒ Drawings *(35 USC 113) Total Sheets* 5
4. Oath or Declaration *Total Pages* 2
   a. ☒ Newly executed (original or copy)
   b. ☐ Copy from a prior application (37 CFR 1.63(d))
   (for continuation/divisional with Box 17 completed)
   [Note Box 5 below]
   i. ☐ DELETION OF INVENTOR(S)
   Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).

5. ☐ Microfiche Computer Program *(Appendix)*

6. ☐ Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. ☐ Computer Readable Copy
   b. ☐ Paper Copy (identical to computer copy)
   c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

7. ☒ Assignment Papers (cover sheet & document(s))
8. ☒ 37 CFR 3.73(b) Statement (when there is an assignee) ☒ Power of Attorney
9. ☐ English Translation Document (if applicable)
10. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
11. ☐ Preliminary Amendment
12. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
13. ☐ Small Entity Statement(s) ☐ Statement filed in Prior Application, Status still proper and desired.
14. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
15. ☐ Other

16. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No:
Prior Application Information: Examiner *Group / Art Unit:*

17. CORRESPONDENCE ADDRESS

☐ *Customer Number or Bar Code Label* or, ☒ *Correspondence Address Below*

Markison & Reckamp, P.C.
P.O. Box 06229
Wacker Drive
Chicago, Illinois 60606-0229
Telephone: 312-939-9800 **Facsimile: 312-939-9828**

Name (Print/Type)	**John R. Garrett**	REGISTRATION NUMBER	**27,888**
Signature	John R Garrett	Date	**August 1, 2000**

FEE TRANSMITTAL

Note: Effective October 1, 1997.
Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT ($) 730.00

Complete if Known

Application Number	
Filing Date	**August 1, 2000**
First Named Inventor	**Mark C. Fowler**
Group Art Unit	
Examiner Name	
Attorney Docket Number	**0100.0001160**

jc784 U.S. PTO 09/629337 08/01/00

METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number	50-0441
Deposit Account Name	**ATI Technologies, Inc.**

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☒ Charge the Issue Fee Set in 37 CFR 1.18 at the mailing of the Notice of Allowance

2. ☐ Payment Enclosed:
☐ Check ☐ Money Order ☐ Other

FEE CALCULATION

1. FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	690	201	345	Utility filing fee	690.00
106	310	206	155	Design filing fee	
107	480	207	240	Plant filing fee	
108	760	208	380	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) ($) 690.00

2. CLAIMS

Claims	Extra	Fee from below	Fee Paid
Total	(-20 =)		
Indep.	(- 3 =)		
Multiple Dep.			

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	78	202	39	Independent claims in excess of 3
104	260	204	130	Multiple dependent claim
109	78	209	39	Reissue independent claims over original patent
110	18	210	9	Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) ______

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	380	216	190	Extension for reply within second month	
117	870	217	435	Extension for reply within third month	
118	1,360	218	680	Extension for reply within fourth month	
128	1,850	228	925	Extension for reply within fifth month	
119	300	219	150	Notice of Appeal	
120	300	220	150	Filing a brief in support of an appeal	
121	260	221	130	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,210	241	605	Petition to revive - unintentional	
142	1,210	242	605	Utility issue fee (or reissue)	
143	430	243	215	Design issue fee	
144	580	244	290	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	40.00
146	690	246	345	Filing a submission after final rejection (37 CFR 1.129(a))	
149	690	249	345	For each additional invention to be examined (37 CFR 1.129(b))	
Other fee (specify)					
Other fee (specify)					

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 40.00

SUBMITTED BY: MARKISON & RECKAMP, P.C.				Complete (if applicable)	
Typed or Printed Name	**John R. Garrett**			Reg. Number	**27,888**
Signature	John R. Garrett	Date	**8/1/00**	Deposit Account User ID	**50-0441**

PATENT APPLICATION
DOCKET NO. 0100.0001160

In the United States Patent and Trademark Office

FILING OF A UNITED STATES PATENT APPLICATION

Title:

OPTIMIZED PRIMITIVE FILLER

Inventor:

Mark C. Fowler **5 Thayer Heights Road** **Hopkinton, MA 01748**	**Kevin M. Olson** **155-11 Broadmeadow Road** **Marlborough, MA 01752**

Attorney of Record
John R. Garrett
Registration No. 27,888
P.O. Box 06229
Wacker Drive
Chicago, Illinois 60606-0229
Phone (312) 939-9800
Fax (312) 939-9828

Express Mail Label No. EL504284757US

Date of Deposit: 8/1/00

I hereby certify that this paper is being deposited with the U.S. Postal Service "Express Mail Post Office to Addresses" service under 37 C.F.R. Section 1.10 on the 'Date of Deposit', indicated above, and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Name of Depositor: Rosalie Swanson
(print or type)

Signature: Rosalie Swanson

OPTIMIZED PRIMITIVE FILLER

Field Of The Invention

The present invention relates to the field of processing graphic images, and, in particular, to graphics rasterization.

Background Of The Invention

In the field of computer graphics, the graphics rendering pipeline is the core of real time graphics. The function of the pipeline is to generate, or render, two dimensional images, three dimensional objects, light sources, lighting models, textures and more. The locations and shapes of the objects in the scene are determined by the geometry, placement of the camera in the environment and the characteristics of that environment. The appearance of the objects is affected by material properties, light sources, textures and lighting models.

The process of rasterizing in computer graphics defines a particular scene in terms of primitives, which are typically triangles. For a particular scene, the area to be displayed by a computer system is termed a screen region. The screen region usually is an area that is less than the total scene. Therefore, some of the primitives lie outside of the screen region, while others lie either partially in the screen region or completely within the screen region. During rasterization the primitives are processed such that pixels contained within the primitives are given values in terms of their color, textures, transparency, etc.

Initially during rasterization, the primitives are filled with a solid color or pattern. The function of filling primitives can be broken into two parts. First a decision must be

made as to which pixels to "fill" (assign values) within a primitive, and also as to what values to assign to these pixels. If the pixels are to be assigned a solid color, for example, then primitives which lie entirely within the screen region have each of their pixels assigned the color value. However, if a primitive lies only partially within the screen area, to assign all the pixels a particular value and then to discard the portion of the primitive lying outside of the screen area requires large amounts of computer processing time and is inefficient.

The process of clipping is the process of determining the portion of a primitive that is within a clipping region, such as the screen region. By clipping the rectangle to the screen region drawing time is saved during the rasterization. Scissoring is the process of computing all points of the primitive, and then drawing only those lines within in the rasterized clip region. Although a number of algorithms are set forth in the prior art with regards to scissoring and clipping, increasing complexity of graphic images has required more efficient methods of clipping or scissoring primitives with regards to the screen region.

Therefore, there is a need in the prior art to provide a more efficient method and apparatus for culling pixels of a primitive that are not in a scissors region. A drawback of the prior art is an implementation of scissoring primitives which fills all the primitives with pixels and then for every pixel has hardware check x,y pixel coordinates against the scissor planes and cull each pixel individually if it is outside the scissor plane. This prior art method is very inefficient in terms of performance as it requires hardware to completely fill all primitives, even for pixels that are outside the scissor plane. Another prior art method is to perform clipping at intersections of primitive edges with the scissor plane. New primitives are then created by subdividing the original primitives in an interactive manner until there are no primitives to intersect the scissor plane. This prior art method of clipping is slow and costly in terms of hardware implementations.

Brief Description Of The Drawings

The features of the present invention which are believed to be novel are set forth with particularity in the appended claims. The invention, together with further objects and advantages, may best be understood by reference to the following description taken in conjunction with the accompanying drawings, and the several figures of which like reference numerals identify like elements.

FIG. 1 depicts a 3D pipeline on a graphics chip according to the present invention.

FIG. 2 is a more detailed block diagram of the raster engine in FIG. 1.

FIG. 3 depicts the location of various primitives relative to a screen region.

FIG. 4 schematically depicts directional values for primitives relative to the screen region.

FIG. 5 is a flow chart of a method of the present invention.

Detailed Description Of a Preferred Embodiment of The Invention

The present invention is used in a computer system, such as a computer system that displays graphic images. The method of the present invention is, in particular, for rasterizing primitives. A first step of the inventive method is to determined if a primitive is totally outside a predetermined screen region. The primitive is discarded if the primitive is totally outside the screen region. If the primitive is not totally outside the screen region, at least a portion of the primitive is identified that lies within the screen region. Then only those pixels in the portion of the primitive that is inside the screen region are filled. These steps are then executed for each primitive of the plurality of primitives that forms the scene of which the screen region is the portion that the computer system displays.

In general terms, a start point is defined at a vertex of a triangular primitive. It is then determined if the start point is outside the screen region. The edge of the primitive is then edge walked from the start point to a boundary of the screen region. A portion of the primitive inside the screen region is span walked and each pixel in the portion of the primitive that is inside the screen region is filled. In a typical graphic system, the

primitive is a triangle and the start point is a vertex of the triangle. Edge walkers and span walkers are known in the art, for example.

FIG. 1 is a block diagram of a graphics chip 100 operatively connected to a frame buffer 104 and receiving a command stream 101. In particular, the command stream 101 is received by graphics processor 103 from other stages in the computer equipment, which are not shown. The command stream contains the information for forming an image on a display. Graphics processor 103 processes the information as known in the art to provide primitives 120 that are representative of the desired image. The primitives 120 are received by a set up engine 108 in a 3D pipeline 102. In the set up engine 108, the x,y coordinates of the primitives 120 are transformed to form screen coordinates. The screen coordinates together with the z coordinate are also referred to as window coordinates. The primitives are then mapped in this window and have x,y coordinates and z coordinates which indicate which primitives are in front of which primitives. These primitives are then passed on to the raster engine 110. The raster engine 110 is also referred to as a scan converter which converts the two dimensional vertices in screen space with at least a z value, a color, and a texture coordinate associated with each vertex into pixels. Unlike previous stages that performed polygon operations, the raster engine stage handles pixel operations. Pixel pipe 112 is operatively connected to the raster engine 110 and a render backend block 114 is operatively connected to the pixel pipe 112. A frame buffer 104, which is connected to the render backend block 114, has at least a color buffer 116 and a z buffer 118. The color buffer 116 stores color information corresponding to pixels in the display frame, and the z buffer 118 stores corresponding z values for the pixels in the display frame.

FIG. 2 shows a block diagram of a graphic primitive clipping system according to the present invention which consists of the set up engine 108 and the raster engine 110. The set up engine 108 receives the primitives 120 and outputs the primitive data 200. The raster engine 110 receives the primitive data 200 in a primitive locator module 202. The primitive locator module 202 supplies at least location values only of primitives that are at least partially within the screen region. These location values are

received by the edge walker module 204, which processes the information and supplies data identifying the portion of the primitive that is inside of the screen region. It should be understood that the primitive locator module 202 identifies primitives which are partially within the screen region and primitives which are completely within the screen region. The edge walker 204 is operatively coupled to the span walker and fill module 206. The span walker and fill module functions to fill, that is, to assign a color value, for example, to each pixel in the portion of the primitive that is within the screen region. Of course, it also assigns values to each of the pixels of a primitive which is totally within the screen region. Raster engine 110 then outputs the pixels 208 for further processing by the 3D pipeline 102.

The present invention does not fill primitives or any portion of primitives that are outside the screen region nor does it implement any hardware or require any hardware to calculate an intersection of primitives with the screen region. Instead, the present invention incorporates the screen region into the edge walking routine, that is, the edge walking module will only start filling the primitive when it is inside the screen region (image region). According to the present invention, if the edge walker starts at a vertex of the primitive outside the screen region, it then travels to the nearest point of primitive intersection with the screen region. The present invention will also terminate filling of a primitive when it transitions from inside of the screen region to outside the screen region.

Depicted, for example, in FIG. 3, is a screen region 300 and a plurality of primitives 302-310. As depicted in FIG. 3, primitive 302 is totally outside of the screen region 300, while the remaining primitives 304-310 are partially within the screen region 300. Each of the primitives 304-310 have different orientations relative to the screen region 300. For example, primitive 304 has its right hand portion within the screen region 300, while primitive 310 has its left hand portion within the screen region 300. The primitive 306 overlaps a corner of the screen region 300, and primitive 308 has its upper portion within the screen region 300.

The screen region 300 is defined by a coordinate system, wherein the bottom left corner of the screen region 300 is denoted as XLEFT, YBOTTOM, and has an upper right hand corner denoted by XRIGHT, YTOP. In this coordinate system, the x direction runs from left to right across the screen region 300, and the y direction runs from bottom to top across the screen region 300. Each of the primitives can be defined relative to the coordinate system and in the present invention, each primitive, such as primitive 302, is defined by a vertex XSTART, YSTART, and a value XEND which is the furthest x direction extent and a value YEND which is the furthest y direction extent from the starting value vertex XSTART, YSTART.

Directional vlaues used by the present invention are depicted in FIG. 4. In FIG. 4 the screen region 300 is shown surrounded by directional symbols. The present invention assigns values to x and y directions of 1 and 0 as follows. An x direction value of 1 indicates right to left, while an x direction value of 0 indicates left to right. A y direction value of 1 indicates bottom to top while a y direction value of 0 indicates top to bottom. FIGs. 3 and 4 will be utilized in explaining the method of the present invention which is depicted in the flow chart of FIG. 5. In a first step 500 of the method of the present invention, a screen region 300 is defined by the screen values XLEFT, XRIGHT, YTOP, YBOTTOM. A primitive to be processed is then defined by XSTART, XEND, YSTART, YEND as shown in FIG. 3 in step 502. For the primitive the values XSTART and XEND define an x direction extent and location of the primitive in the coordinate system. Similarly, the values YSTART and YEND define a y direction extent and location of the primitive in the coordinate system. For the screen region 300, the values XLEFT and XRIGHT define an x direction extent and location of the screen region in the coordinate system, while YTOP and YBOTTOM define a y direction extent and location for the screen region in the coordinate system. In step 503 the variables x and y for the primitive are initially set to the XSTART, YSTART values.

In step 504 it is determined if the primitive is totally outside of the screen region 300, such as a primitive 302 in FIG. 3. The primitive 302 will be found to be totally

outside the screen region 300 if at least one of the following is logically true given a start point of x = XSTART and y = YSTART for the primitive:

XDIR AND ((X<XLEFT) OR (XEND>XRIGHT))

XDIR AND ((X>XRIGHT) OR (XEND>XLEFT))

YDIR AND ((Y<YTOP) OR (YEND>YBOTTOM))

YDIR AND ((Y>YBOTTOM) OR (YEND<YTOP)).

If the primitive is found to be totally outside of the screen region, such as primitive 302 in FIG. 3, the primitive is then discarded without filling any pixels therein in step 506 of the method depicted in FIG. 5. If the primitive is not totally outside of the screen region 300, then it must be determined what portion of the primitive lies within the screen area 300.

The portion of the primitive lying within the screen area 300 is identified as follows. First, the variable y is incremented if the following first value is logically true:

(((YDIR AND (Y > YBOTTOM)) OR ((YDIR AND (Y < YTOP)))

Then the variable x is incremented if the following second value is logically true:

(((XFIR AND X>XRIGHT)) OR ((XDIR AND (X<XLEFT))).

These steps (508, 510) are then repeated until the first and second values are not true, which identifies a beginning of a portion of the primitive that is inside the screen region. That is, the test is to determine if the variable y is between YTOP and YBOTTOM, and if the variable x is between XLEF and XRIGHT. If the answer to this question is no, then x and y are incremented. If the answer to the question is yes, then the pixel corresponding to that x,y coordinate lies within the portion of the primitive within the screen region 300, and the pixel is then filled (step 512), that is, assigned the proper

color value. The process ends when all pixels within the portion of the primitive inside the screen region 300 have been filled, the test being performed in step 514 in FIG. 5.

It is to be noted that in general the method steps and the equations involve comparisons of the current x and y coordinates to the XLEFT, XRIGHT and YTOP and YBOTTOM of the screen area, as well as the X direction and Y direction values at that x,y coordinate. Note that when x and y are incremented, there are eight possible directions of movement as depicted in FIG. 4. For example, the direction 401 has an X direction from left to right and is assigned a value of 0, and a Y direction from bottom to top which has an assigned value of 1. The direction symbol 402 has an X direction of left to right and therefore has an assigned value of 0 and a Y direction of top to bottom which has an assigned value of 0. Direction symbol 403 would have an X direction value of 0 and a Y direction value of 1. The direction symbol 404 has an X direction value of 0 and a Y direction value of 1. The direction symbol 405 has an X direction value of 0 and a Y direction value of 1. The direction symbol 406 has an X direction value of 0 and a Y direction value of 1. The direction value symbol 407 has an X direction value of 0 and a Y direction value of 0. The direction symbol 408 has an X direction value of 0 and a Y direction value of 0.

The primitive location module 202 in one embodiment is software that implements the step of determining if the primitive is totally outside the screen region. Similarly, the edge walker module 204 and the span walker module 206 are software implementations for performing the above described steps.

When the edge walker 204 in the raster engine 110 begins at the XSTART, Y START values of a primitive, it walks one of the edges of the primitive until it reaches the closest boundary point of intersection of the primitive with the screen region. For example, as shown in FIG. 3, the edge walker would begin in primitive 304 at the vertex 320 and proceed along edge 322 until it reaches the point of intersection 324. The dotted line 326 schematically depicts how the edge walker proceeds in the X direction and in the Y direction until it arrives at the intersection point 324. In reality, it is to be understood that the actual movement is a stair step-type movement along the edge 322.

For primitive 360, the edge walker starts at vertex 330 and proceeds until it identifies point 332 as being the first point of intersection between the primitive 306 and the screen region 300. It is to be understood that once the area within the screen region 300 is identified, the span walker and fill module 206 then take over and fill each of the pixels and the portion of the primitive which is within the screen region 300. This is indicated by the area of the primitives which are filled with lines in FIG. 3. Primitive 308 shows that the edge walker would start at the vertex 340 and proceed to the intersection point 342 at which time the span walker fill module would take over to fill a portion of the primitive 308 which is within the screen region 300. Finally, it is shown how the method of the present invention also works for primitive 310 wherein the start point 350 is within the screen region 300. The equations then cause the span walker and fill module 206 to fill the portion of the primitive 310 which is within the screen region 300 but not the portion of the primitive 310 which is outside of the screen region 300.

Thus, in general, in the primitive location module 202 the primitive values are compared to the screen region values to determine if the primitive is totally outside the screen region. If this is not true, then the edge walker module 204 and the span walker and fill module 206 determine the start point of the intersection of the primitive in the screen region for filling the pixels that are within this portion. The filling of the pixels within the portion within the screen region is finished when one of the following is true:

(XDIR AND (X < XLEFT)),

(XDIR AND (X > XRIGHT)),

(YDIR AND (Y < YTOP)),

(YDIR AND (Y > YBOTTOM)).

Thus, the invention fulfills the need in the prior art for an efficient method of filling the portions of primitives lying within a screen region. The method of the present invention in particular does not fill the portions of the primitives outside of the screen region, thus saving significant computing time and resources. Furthermore, the method of the present invention does not require a hardware calculation of the intersection of the primitives with the screen region as is done in the prior art.

The invention is not limited to particular details of the apparatus and method depicted and the modifications and applications may be contemplated. Certain other changes may be made in the above-described method and apparatus without departing from the true spirit of the scope of the invention herein involved. For example, the method of the present invention can be utilized in other computer systems other than the 3D pipeline embodiment depicted in FIG. 1. Furthermore, the primitives can have other

configurations than triangular, and the starting point can occur at other locations on the primitive rather than the vertex of a triangle. It is intended, therefore, that the subject matter in the above depiction should be interpreted as illustrative and not in a limiting sense.

Claims

WHAT IS CLAIMED IS:

1. In a computer system, a method for rasterizing primitives, comprising the steps of:

determining if a primitive is totally outside a predetermined screen region or at least partially within the predetermined screen region;

discarding the primitive if the primitive is totally outside the screen region;

finding at least a portion of the primitive that is inside the screen region if the primitive is not totally outside the screen region;

filling only pixels in the portion of the primitive that is inside the screen region.

2. The method according to claim 1, wherein the method further comprises repeating the method steps for each primitive of a plurality of primitives.

3. The method according to claim 1, wherein in the step of determining if the primitive is inside or outside of the screen region, the method further comprises the steps of:

providing an X,Y coordinate system:

determining values of XSTART, YSTART, XEND, YEND for the primitive, XSTART and XEND defining an X direction extent and location of the primitive in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the primitive in the coordinate system;

providing values of XLEFT, XRIGHT, YTOP, YBOTTOM for the screen region, XLEFT and XRIGHT defining an X direction extent and location of the screen region in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the screen region in the coordinate system; and

comparing the primitive values to the screen region values to determine if the primitive is totally outside the screen region.

4. The method according to claim 3, wherein, the method further comprises the steps of:

defining first and second x direction values of 0 and 1, respectively, for an x direction XDIR in the coordinate system as, respectively, left to right and right to left relative to the screen region, and defining first and second y direction values as 0 and 1, respectively, for a y direction YDIR in the coordinate system as, respectively, top to bottom and bottom to top;

determining that the primitive is totally outside the screen area if at least one of the following is logically true given a start point X=XSTART and Y=YSTART for the primitive:

XDIR AND ((X<XLEFT) OR (XEND>XRIGHT))

XDIR AND ((X>XRIGHT) OR (XEND>XLEFT))

YDIR AND ((Y<YTOP) OR (YEND>YBOTTOM))

YDIR AND ((Y>YBOTTOM) OR (YEND<YTOP)).

5. The method according to claim 3, wherein in the step of finding at least a portion of the primitive that is inside the primitive, given a start point X=XSTART and Y=YSTART for the primitive, the method further comprises the steps of:

(1) incrementing Y if a first value, (((YDIR AND (Y>YBOTTOM)) OR ((<u>YDIR</u> AND (Y<YTOP))), is logically true;

(2) incrementing X if a second value, (((XDIR AND (X>XRIGHT)) OR ((<u>XDIR</u> AND (X<XLEFT))), is logically true; and

(3) repeating steps (1) and (2) until the first and second values are not true, which identifies a beginning of a portion of the primitive that is inside of the screen region.

6. The method according to claim 3, wherein in the step of filling the filling is finished when one of the following is true:

(XDIR AND (X < XLEFT)),

(<u>XDIR</u> AND (X > XRIGHT)),

(YDIR AND (Y < YTOP)),

(<u>YDIR</u> AND (Y > YBOTTOM)).

7, The method according to claim 1, wherein the method further comprises the steps of:

defining a start point on an edge of the primitive;

determining if the start point is outside the screen region;

edge walking the edge of the primitive from the start point to a boundary of the screen region;

span walking a portion of the primitive inside the screen region and filling each pixel in the portion of the primitive that is inside the screen region.

8. The method according to claim 7, wherein the primitive is a triangle and the start point is a vertex of the triangle.

9. The method according to claim 1, wherein the primitive is a triangle.

10. A graphic primitive clipping system that receives primitives and clips the primitives relative to a predetermined screen region, comprising:

a setup engine having an input for receiving a primitive and an output for supplying at least location values of the primitive relative to the screen region;

a primitive locator module having an input operatively connected to the output of the setup engine and having an output for supplying the at least location values only of primitives that are at least partially within the screen region;

an edge walker module having an input operatively connected to the output of the primitive locator module and having an output for supplying data identifying the portion of the primitive inside of the screen region;

a span walker having an input operatively connected to the output of the edge walker and an output for supplying filled pixels for pixels in the portion of the primitive inside of the screen region

11. The system according to claim 10, wherein the system processes each primitive of a plurality of primitives.

12. The system according to claim 10, wherein the primitive locator module compares the location values of the primitive to starting and ending values of the screen region values to determine if the primitive is totally outside the screen region.

13. The system according to claim 12, wherein the primitive and the screen region lie in an X,Y coordinate system, wherein the location values of the primitive are XSTART,

YSTART, XEND, YEND, XSTART and XEND defining an X direction extent and location of the primitive in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the primitive in the coordinate system, and wherein the screen region has limit values of XLEFT, XRIGHT, YTOP, YBOTTOM, XLEFT and XRIGHT defining an X direction extent and location of the screen region in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the screen region in the coordinate system, wherein the primitive further is defined by first and second x direction values of 0 and 1, respectively, for an x direction XDIR in the coordinate system as, respectively, left to right and right to left relative to the screen region, and first and second y direction values as 0 and 1, respectively, for a y direction YDIR in the coordinate system as, respectively, top to bottom and bottom to top, and wherein the primitive is totally outside the screen area if at least one of the following is logically true given a start point X=XSTART and Y=YSTART for the primitive:

$\text{XDIR AND ((X} < \text{XLEFT) OR (XEND} > \text{XRIGHT))}$

$\underline{\text{XDIR}}\text{ AND ((X} > \text{XRIGHT) OR (XEND} > \text{XLEFT))}$

$\text{YDIR AND ((Y} < \text{YTOP) OR (YEND} > \text{YBOTTOM))}$

$\underline{\text{YDIR}}\text{ AND ((Y} > \text{YBOTTOM) OR (YEND} < \text{YTOP)).}$

14. The method according to claim 13, wherein the edge walker module finds at least a portion of the primitive that is inside the primitive, given a start point X=XSTART and Y=YSTART for the primitive, by:

(1) incrementing Y if a first value, $\text{(((YDIR AND (Y} > \text{YBOTTOM)) OR ((}\underline{\text{YDIR}}\text{ AND (Y} < \text{YTOP))),}$ is logically true;

(2) incrementing X if a second value, (((XDIR AND (X>XRIGHT)) OR (($\underline{\text{XDIR}}$ AND (X<XLEFT))), is logically true; and

(3) repeating (1) and (2) until the first and second values are not true, which identifies a beginning of a portion of the primitive that is inside of the screen region.

15. The system according to claim 13, wherein the span walker has filled all pixels in the portion of the primitive inside of the screen region when one of the following is true:

(XDIR AND (X < XLEFT)),

($\underline{\text{XDIR}}$ AND (X > XRIGHT)),

(YDIR AND (Y < YTOP)),

($\underline{\text{YDIR}}$ AND (Y > YBOTTOM)).

15. The method according to claim 13, wherein the primitive is a triangle and the start point is a vertex of the triangle.

16. The method according to claim 10, wherein the primitive is a triangle.

17. A graphic primitive clipping system that clips triangular primitives relative to a predetermined screen region, each primitive defined by location values XSTART, YSTART, XEND, YEND, XSTART and XEND defining an X direction extent and location of the primitive in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the primitive in the coordinate system, a screen region defined by limit values XLEFT, XRIGHT, YTOP, YBOTTOM, XLEFT and XRIGHT defining an X direction extent and location of the screen region in the coordinate system, and YSTART and YEND defining a Y direction extent and location of the screen region in the coordinate system , the primitive further defined by first and second x direction values of 0 and 1, respectively, for an x direction XDIR in the coordinate system as, respectively, left to right and right to left relative to the screen region, and first and second y direction values as 0 and 1, respectively, for a y direction YDIR in the coordinate system as, respectively, top to bottom and bottom to top, comprising:

a primitive locator module having an input for receiving primitives and having an output for supplying only primitives that are at least partially within the screen region, primitives being totally outside the screen area if at least one of the following is logically true given a start point X=XSTART and Y=YSTART for a primitive

XDIR AND ((X<XLEFT) OR (XEND>XRIGHT))

$\underline{\text{XDIR}}$ AND ((X>XRIGHT) OR (XEND>XLEFT))

YDIR AND ((Y<YTOP) OR (YEND>YBOTTOM))

$\underline{\text{YDIR}}$ AND ((Y>YBOTTOM) OR (YEND<YTOP));

an edge walker module having an input operatively connected to the output of the primitive locator module and having an output for supplying data identifying the portion of the primitive inside of the primitive, the edge walker module structured such that at least a portion of the primitive that is inside the primitive, given a start point X=XSTART and Y=YSTART for the primitive, being found by:

(1) incrementing Y if a first value, (((YDIR AND (Y>YBOTTOM)) OR (($\underline{\text{YDIR}}$ AND (Y<YTOP))), is logically true;

(2) incrementing X if a second value, (((XDIR AND (X>XRIGHT)) OR (($\underline{\text{XDIR}}$ AND (X<XLEFT))), is logically true; and

(3) repeating steps (1) and (2) until the first and second values are not true, which identifies a beginning of a portion of the primitive that is inside of the screen region.;

a span walker having an input operatively connected to the output of the edge walker and an output for supplying filled pixels for pixels in the portion of the primitive inside of the screen region, the span walker having filled all pixels in the portion of the primitive inside of the screen region when one of the following is true:

(XDIR AND (X < XLEFT)),

($\underline{\text{XDIR}}$ AND (X > XRIGHT)),

(YDIR AND (Y < YTOP)),

($\underline{\text{YDIR}}$ AND (Y > YBOTTOM)).

18. The system according to claim 17, wherein the start point is a vertex of the primitive.

OPTIMIZED PRIMITIVE FILLER

Abstract Of The Invention

The optimized primitive filler is used in a computer system, such as a computer system that displays graphic images. A first step of the method it is determined if a primitive is totally outside a predetermined screen region or at least partially within the predetermined screen region. The primitive is then discarded if the primitive is totally outside the screen region. If the primitive is not totally outside the screen region, at least a portion of the primitive is identified that lies within the screen region. Then only those pixels in the portion of the primitive that is inside the screen region are filled. These steps are executed for each primitive of a plurality of primitives that forms a scene of which the screen region is the portion that the computer system displays. No pixels are filled in primitives which are totally outside the screen region, and no pixels are filled in portions of primitives that are outside the screen region. The optimized primitive filler and its associated method provide a fast and efficient rasterization of primitives.

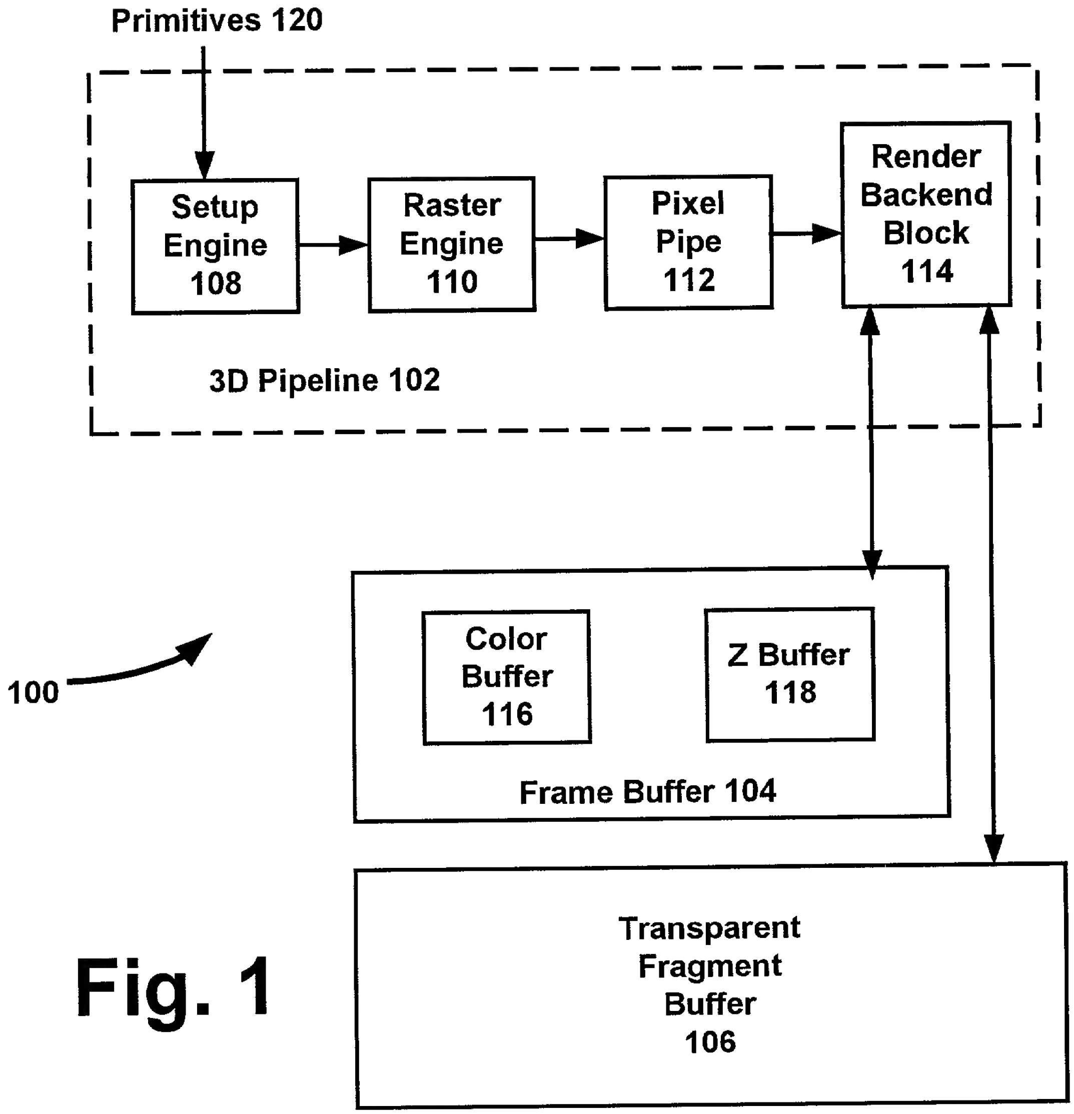

Primitives 120
Setup Engine 108
Raster Engine 110
Pixel Pipe 112
Render Backend Block 114
3D Pipeline 102
100
Color Buffer 116
Z Buffer 118
Frame Buffer 104
Transparent Fragment Buffer 106

Fig. 1

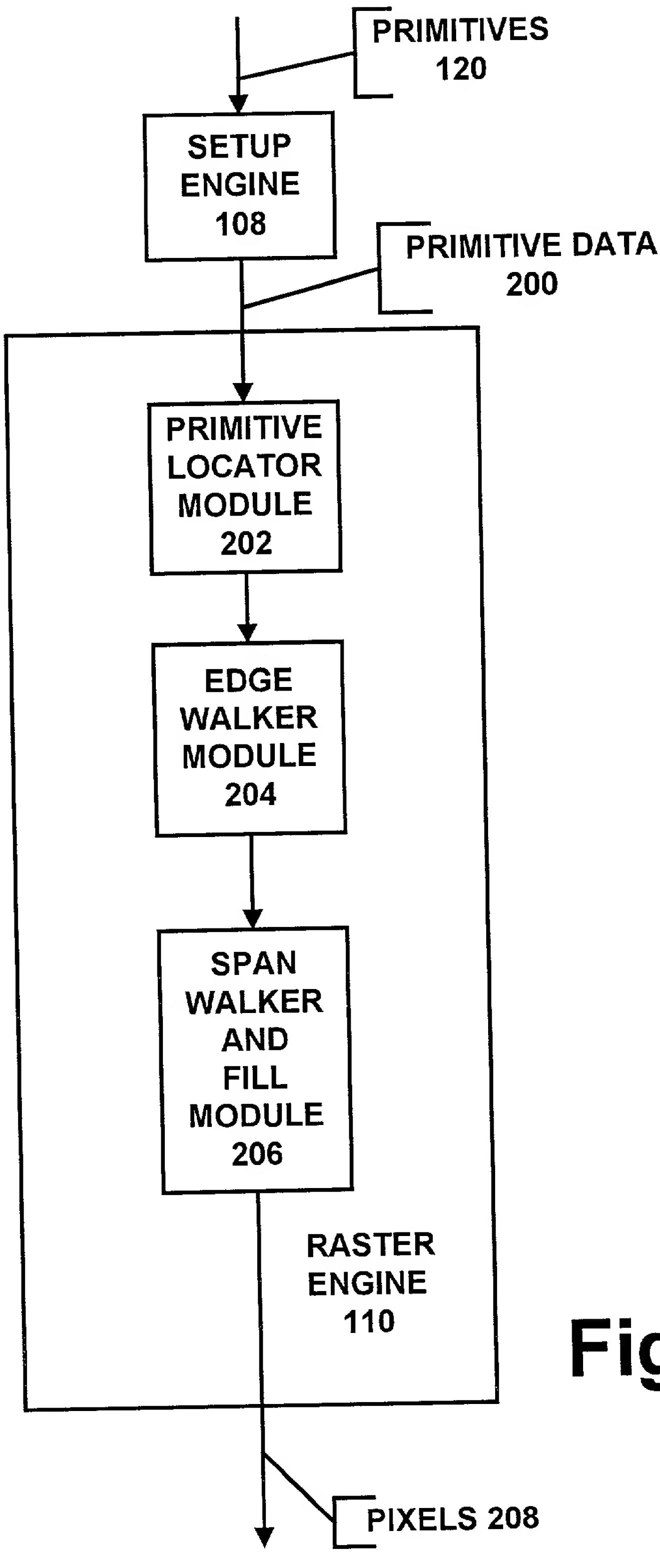
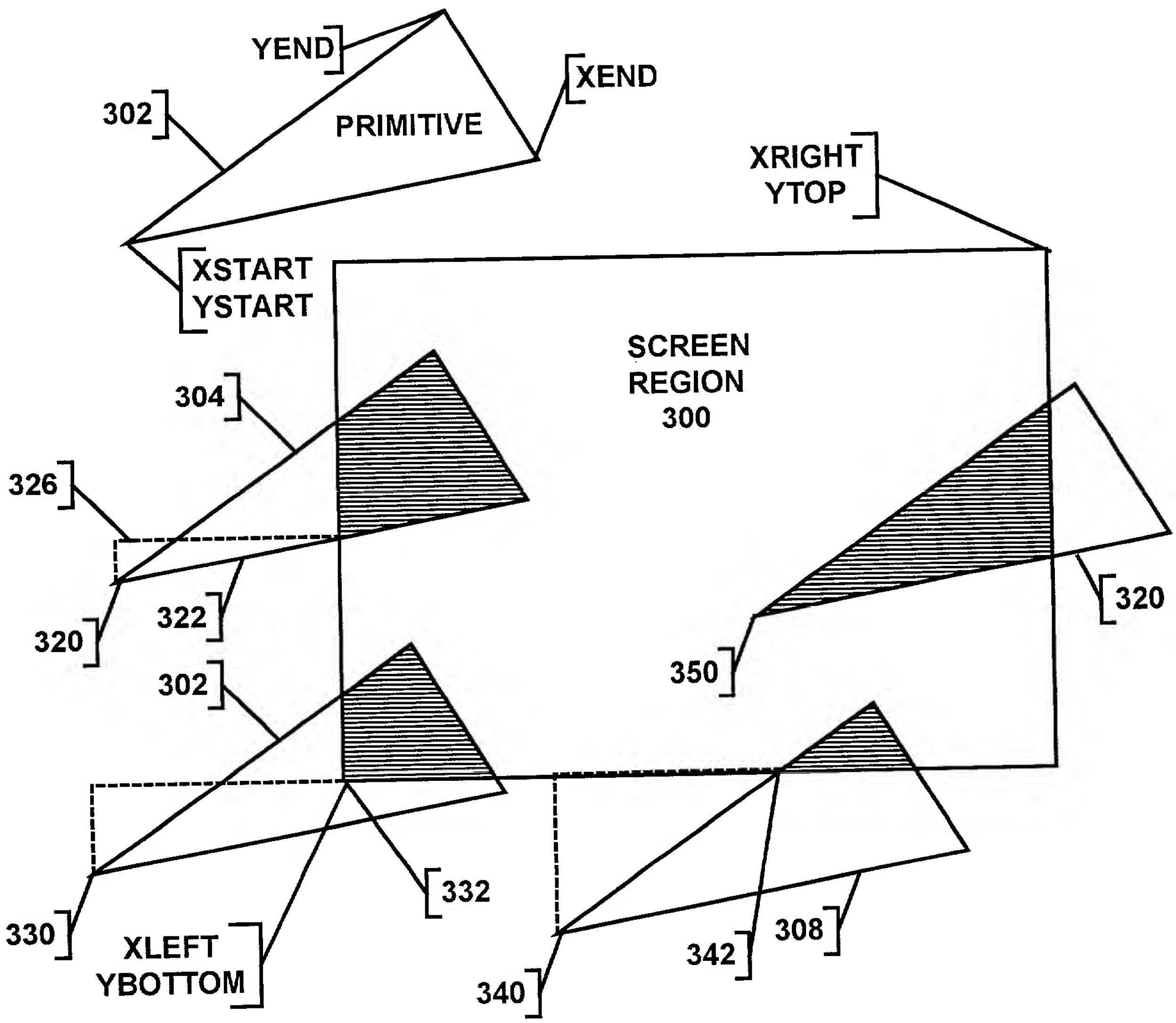

PRIMITIVES
120
SETUP
ENGINE
108
PRIMITIVE DATA
200
PRIMITIVE
LOCATOR
MODULE
202
EDGE
WALKER
MODULE
204
SPAN
WALKER
AND
FILL
MODULE
206
RASTER
ENGINE
110
PIXELS 208

Fig. 2



YEND
XEND
302
PRIMITIVE
XRIGHT
YTOP
XSTART
YSTART
SCREEN
REGION
300
304
326
320
322
320
350
302
332
330
XLEFT
YBOTTOM
342
308
340

Fig. 3

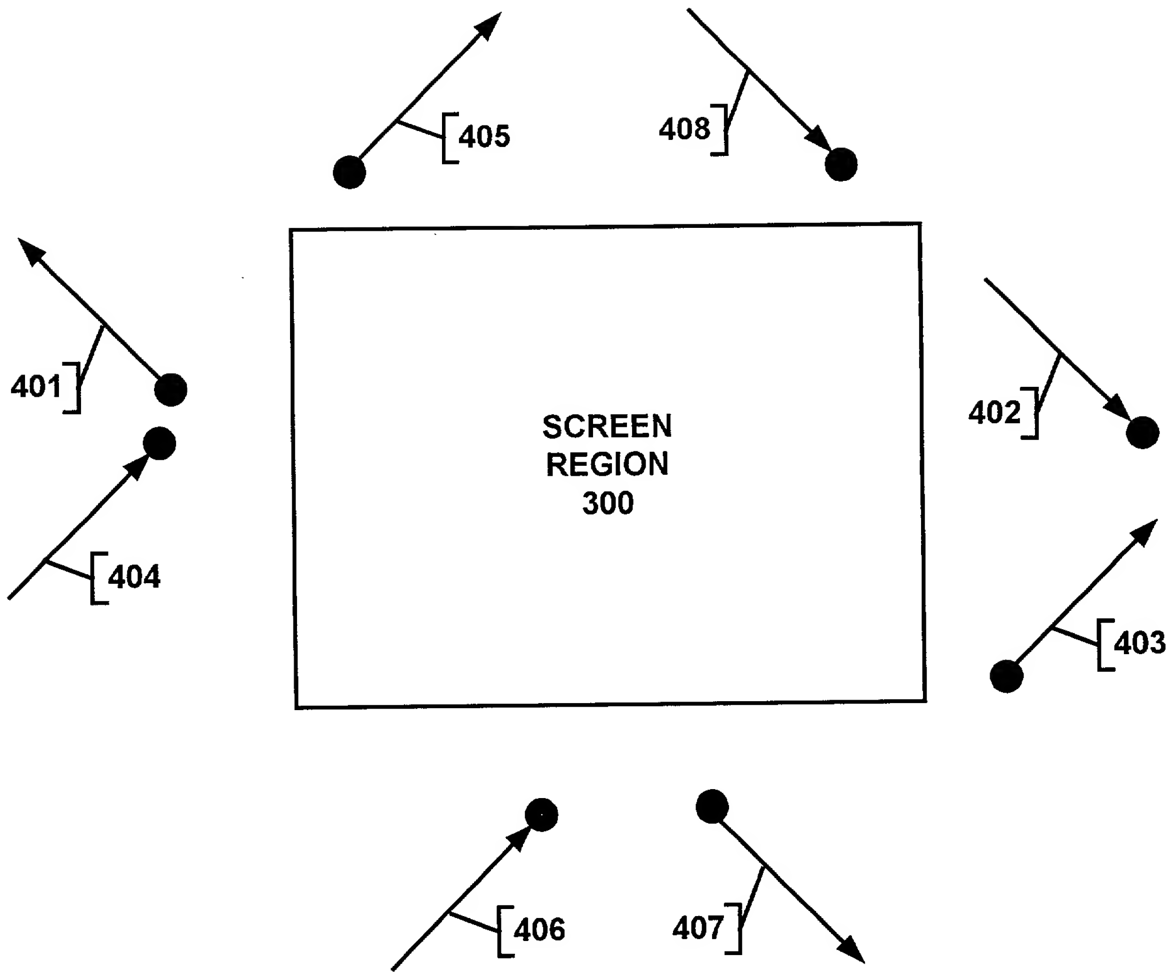


Fig. 4

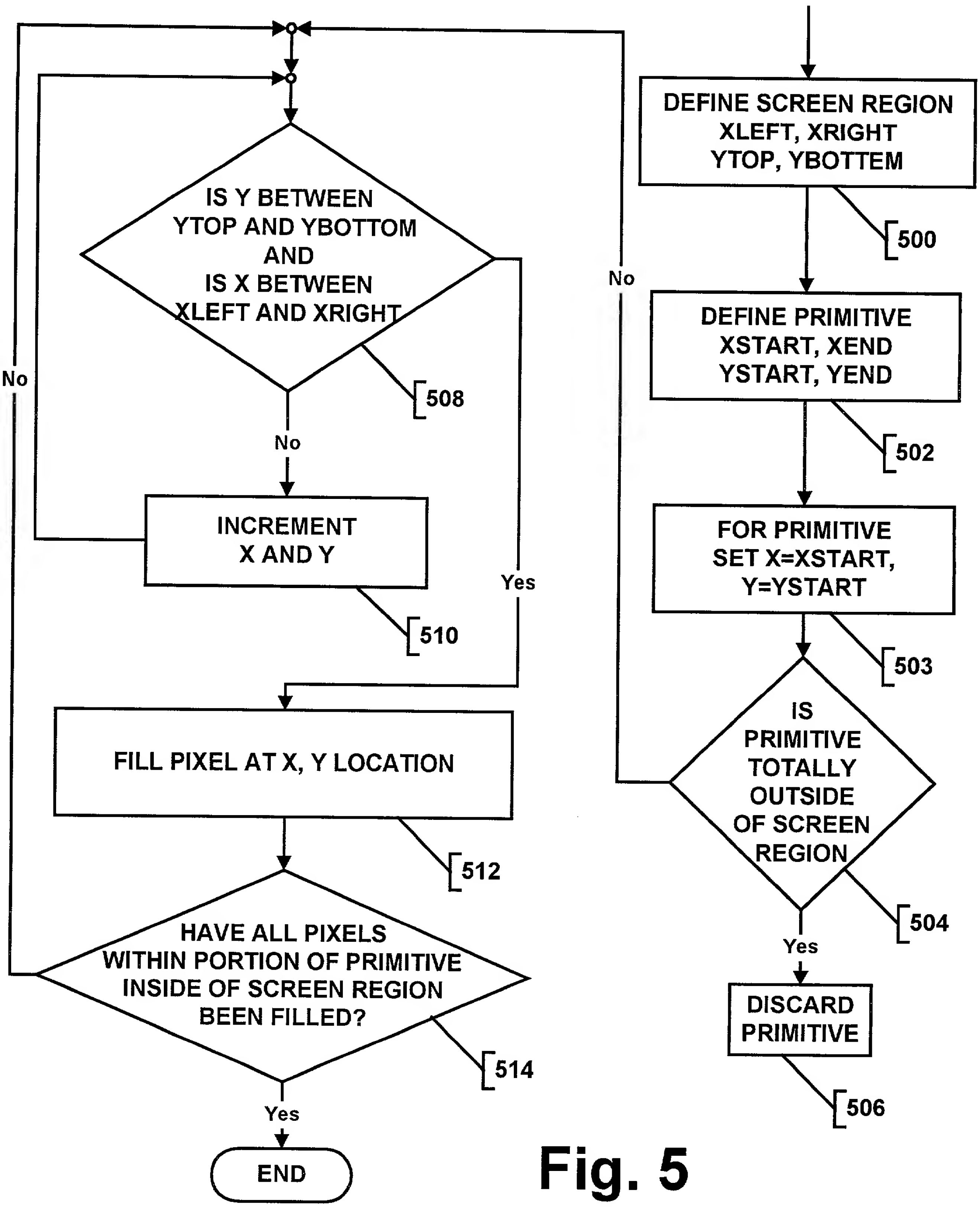

DEFINE SCREEN REGION XLEFT, XRIGHT YTOP, YBOTTEM
500
DEFINE PRIMITIVE XSTART, XEND YSTART, YEND
502
FOR PRIMITIVE SET X=XSTART, Y=YSTART
503
IS PRIMITIVE TOTALLY OUTSIDE OF SCREEN REGION
504
Yes
DISCARD PRIMITIVE
506
No
IS Y BETWEEN YTOP AND YBOTTOM AND IS X BETWEEN XLEFT AND XRIGHT
508
No
INCREMENT X AND Y
510
Yes
FILL PIXEL AT X, Y LOCATION
512
HAVE ALL PIXELS WITHIN PORTION OF PRIMITIVE INSIDE OF SCREEN REGION BEEN FILLED?
514
No
Yes
END

Fig. 5

DECLARATION FOR UTILITY OR DESIGN PATENT APPLICATION (37 CFR 1.63)

☒ Declaration Submitted with Initial Filing, OR
☐ Declaration Submitted after Initial Filing
(surcharge (37 CFR 1.16 (e)) required)

Attorney Docket Number 0100.0001160
First Named Inventor Mark C. Fowler
COMPLETE IF KNOWN
Application Number
Filing Date
Group Art Unit
Examiner Name

As a below named inventor, I hereby declare that:
My residence, post office address, and citizenship are as stated below next to my name.
I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled: **Optimized Primitive Filler**
the specification of which:
☒ is attached hereto.
☐ was file on (MM/DD/YYYY) as United States Application Number or PCT International Application Number and was amended on (MM/DD/YYYY) (if applicable).

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment specifically referred to above.
I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56.

I hereby claim foreign priority benefits under 35 U.S.C. 119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or of any PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application Number(s)	Country	Foreign Filing Date (MM/DD/YYYY)	Priority Not Claimed	Certified Copy Attached? YES	NO
			☐	☐	☐
			☐	☐	☐

☐ Additional foreign application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

I hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below.

Application Number(s)	Filing Data (MM/DD/YYYY)

☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

U.S. Parent Application or PCT Parent Number	Parent Filing Date (MM/DD/YYYY)	Parent Patent Number (if applicable)

☐ Additional U.S. or PCT international application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.

As a named inventor, I hereby appoint the following registered practitioner(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith:

Name	Registration Number	Name	Registration Number
John R. Garrett	27,888	Christopher J. Reckamp	34,414
Paul M. Anderson	39,896		
Sally Daub	41,478		

☐ Additional registered practitioner(s) named on supplemental Registered Practitioner Information sheet PTO/SB/02C attached hereto.

Direct all correspondence to:

Markison & Reckamp, P.C.
P.O. Box 06229
Wacker Drive
Chicago, Illinois 60606-0229
Telephone:312-939-9800
Facsimile: 312-939-9828

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Name of Sole or First Inventor: ☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any])		Family Name or Surname	
Mark C.		**Fowler**	
Inventor's Signature	Mark C. Fowler	**Date** 7/31/2000	
Residence	**City: Hopkinton**	**State: MA**	**Country: USA** / **Citizenship: USA**
Post Office Address	**5 Thayer Heights Road**		
City: Hopkinton	**State: MA**	**ZIP: 01748**	**Country: USA**

Name of Additional Joint Inventor: ☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any])		Family Name or Surname	
Kevin M.		**Olson**	
Inventor's Signature	Kevin M. Olson	**Date** 7/28/2000	
Residence	**City: Marlborough**	**State: MA**	**Country: USA** / **Citizenship: USA**
Post Office Address	**155-11 Broadmeadow Road**		
City: Marlborough	**State: MA**	**ZIP: 01752**	**Country: USA**

Name of Additional Joint Inventor: ☐ A petition has been filed for this unsigned inventor

Given Name (first and middle [if any])		Family Name or Surname	
Inventor's Signature		**Date**	
Residence	**City:**	**State:**	**Country:** / **Citizenship:**
Post Office Address			
City:	**State:**	**ZIP:**	**Country:**

☐ Additional inventors are being named on the ______supplemental Additional Inventor(s) sheet(s) PTO/SB/02A attached hereto.